FOS System Architecture

Andy Miller

16 October 1995

FOS CDR Roadmap

[redacted]

FOS CDR Overview
- FOS CDR goals
- Driving requirements

Engineering Activities
- Activities since PDR
- FOS team approach

[redacted]

[redacted]

IST
- Capabilities
- Plans

Hardware Design
- Computers
- Peripherals

Network Design
- EOC LAN
- IST Connectivity

FOS Infrastructure
- Mgt Services
- Comm Services

Segment Scenarios
- End-to-End Flow
- Subsystem Interfaces
- Building block linkage

[redacted]

Subsystem Design
- Detailed design
- FOS functions/tools
- Subsystem design features

RMA
- RMA allocation
- FMEA/CIL

[redacted]

Operations Overview
- EOC facilities
- FOT positions

Operational Scenarios
- End-to-end flow
- Operations perspective
- FOT tool usage

[redacted]

Development
- Release Plan
- Development approach

Testing
- Test approach
- Test organization

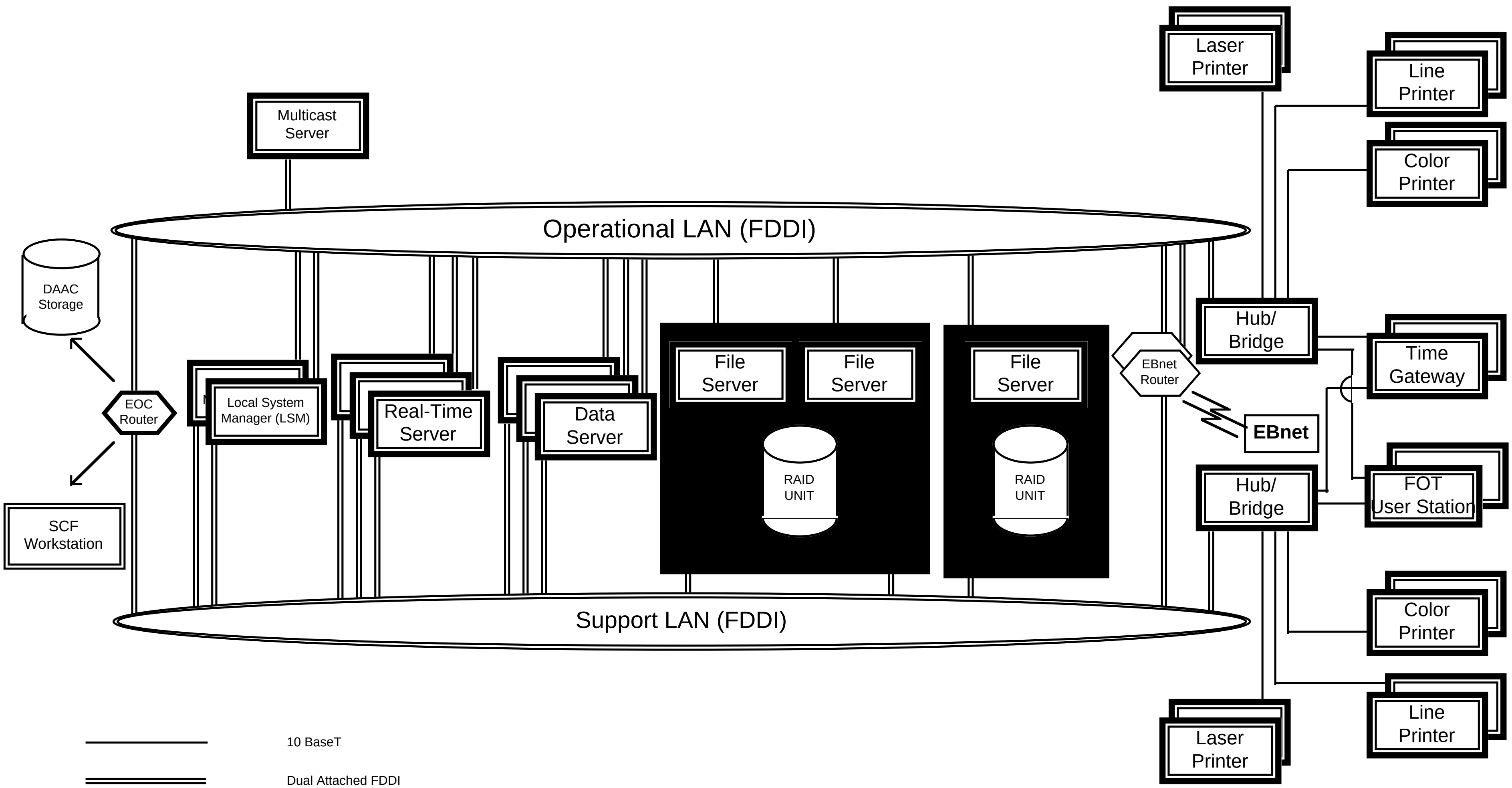
FOS System Architecture

Physical
- Network
- Computers

FOS Infrastructure
- Mgt Services
- Comm Services

Software
- Scheduling
- Real-Time
- Analysis

EOC Block Diagram



FOS Physical Architecture

EOC Network

- **Dual FDDI rings**
  - **Operational LAN**
  - **Support LAN**
- **Ethernet segments**
  - **User Stations connected to EOC LAN via Ethernet segments**

Computers

- **Real-Time Server**
  - **Performs real-time processing of telemetry and command in support of a spacecraft contact**
- **Data Server**
  - **Services requests for scheduling data, historical data, and database information**

FOS Physical Architecture

- User Stations
  - Provides user capability to access all FOS services
- Local System Manager
  - Hosts CSMS Management and Communication Services
- Multicast Replicator Server
  - Provides localized function for routing multicast data to ISTs via point-to-point interface
- Data Storage Unit
  - Provides fault tolerant access to FOS operational data

FOS System Architecture

Real-Time Server

Real-Time Contact
- Establish Logical string
-Modes
 - Real-Time
 - Simulation
 - Replay
- Telemetry
 - Comprehensive TLM Monitor
 - Archive
-Command
 - Validation
 - Build
- Transmission
- Verification

Data Server

Data Management
-Back Orbit Ingest & Archive
-Telemetry Merge
-Archive Events
-PDB Management

Scheduling
- Resource Model

Analysis
-Statistics Generation
-Standing Orders

Data Storage Unit

Scheduling
 - Plans, schedules
 - Orbit Data
 - Loads
 - Ground Scripts
Real-Time
 - Telemetry
 - Command Loads
Analysis
- Statistics
 - Telemetry
 - Ground telemetry
Data Management
 - Telemetry DB
 - Command DB
 - Activities DB
 - Constraints DB

FOT User Station

Scheduling
- Scheduler
- Visualization
- Resource Model

Real-Time
- Monitor real-time contacts
- Modes
  - Simulation
  - Real-Time

- Displays, plts, graphs
- Command
- Execute ground scripts
- Send real-time commands

Off-Line
- Algorithms
- Trending, Fault detection
- Replay
- Analysis Request Manager

Data
- Telemetry
  - Spacecraft
  - Ground
- Statistics

Products
- Reports
- Plots
- Carry-out

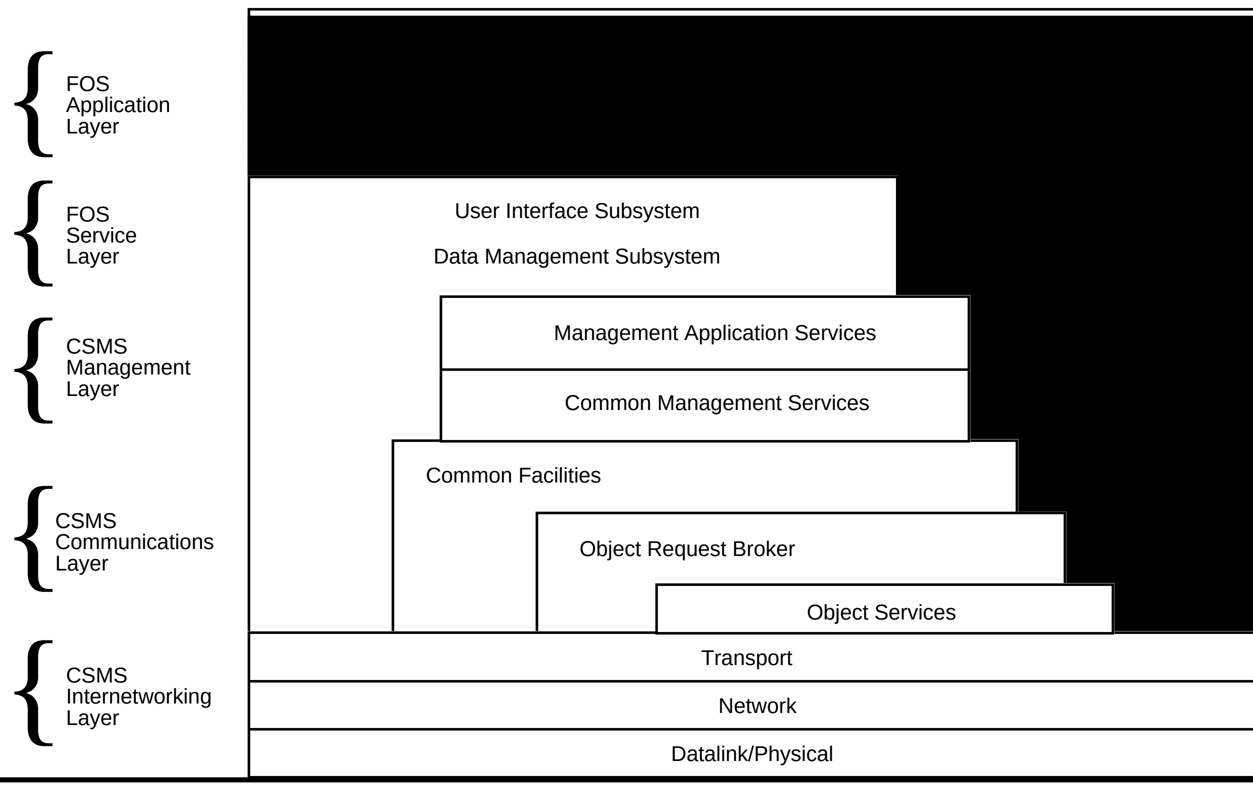
PI/TL Workstation (IST Toolkit)

Scheduling
-Scheduler
-Visualization
- Resource Model

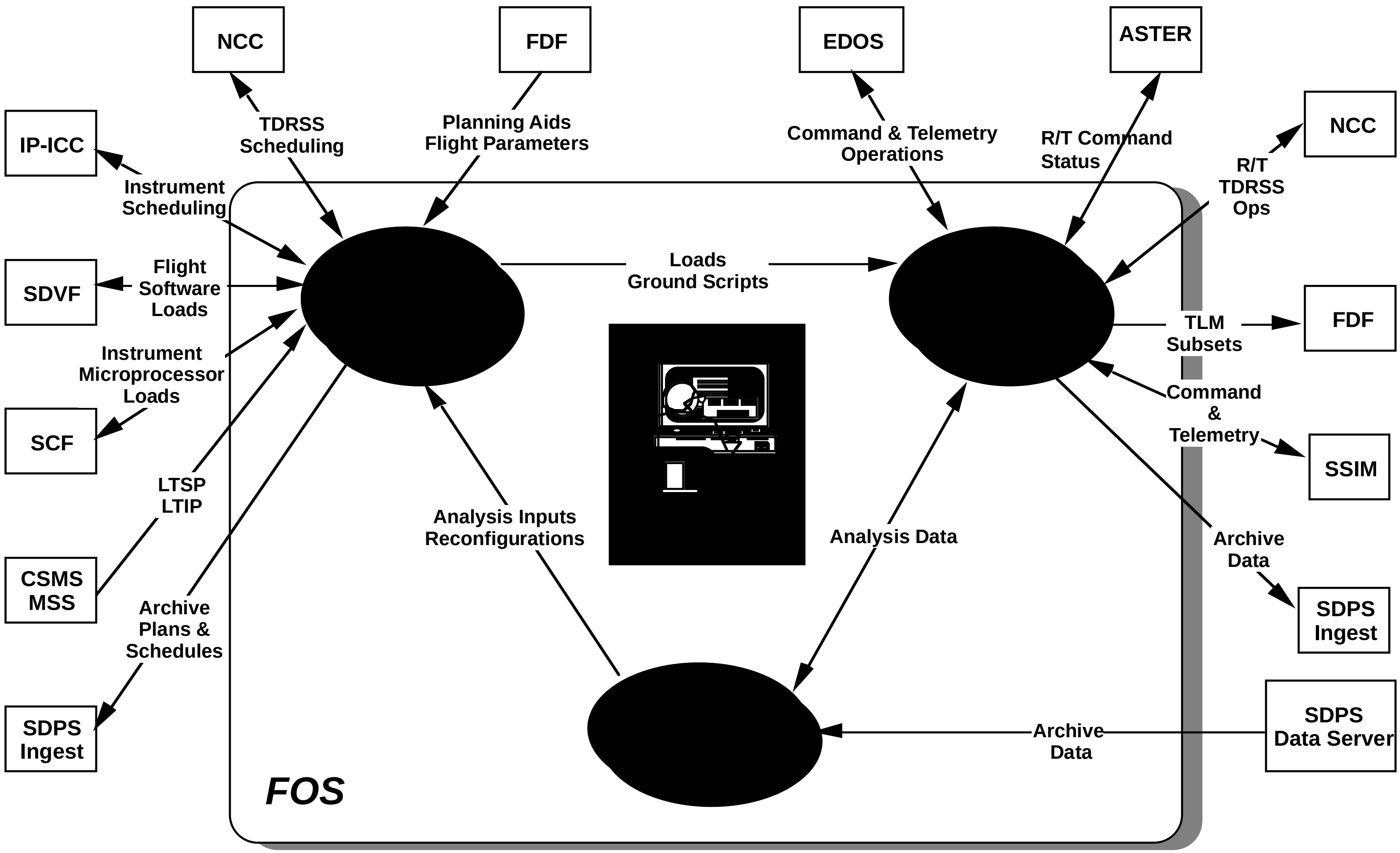
Real-Time
-Monitor Real-Time Contacts
-Send Command Requests

Off-Line
- Same as FOT W/S
  (Analysis Phase)

FOS Software Architecture



FOS Software Context Diagram



Key FOS Threads

Scheduling

- Ingest and distribution of planning aids
- Establishment of TDRSS contact times
- Final scheduling
- Command load generation and ground script generation

Real-Time

- NCC and EOC configuration requests
- Command uplink and verification
- Telemetry processing and monitoring
- Real-Time analysis

Off-Line

- Anomaly detection
- Performance assessment
- Load management

FOS Software Functionality

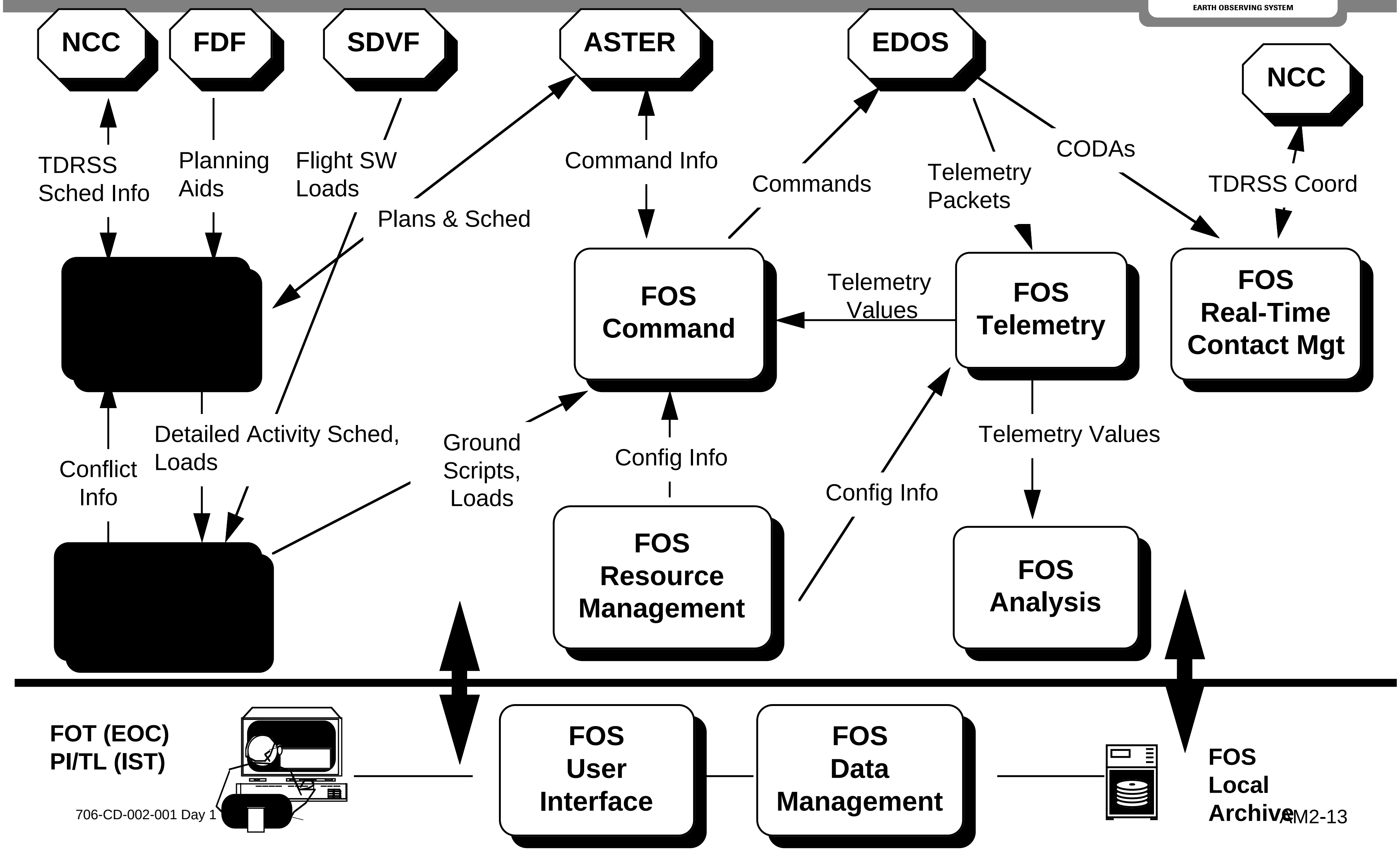
Function to Software Subsystem Mapping

Activity Phase	Software Subsystem
Scheduling	Planning & Scheduling Command Management
Real-time	Resource Management Real-time Contact Management Telemetry Command Analysis
Off-Line	Analysis Command Management

Support	User Interface Data Management

page intentionally left blank.

Scheduling Architecture



Scheduling Architecture

Ingest and Distribution of Planning Aids

- Receive planning aids from FDF
- Distribute to instrument teams via IST

Establishment of TDRSS Contact Times

- TDRSS contact request submitted to NCC
- NCC sends contact schedule with any rejection info

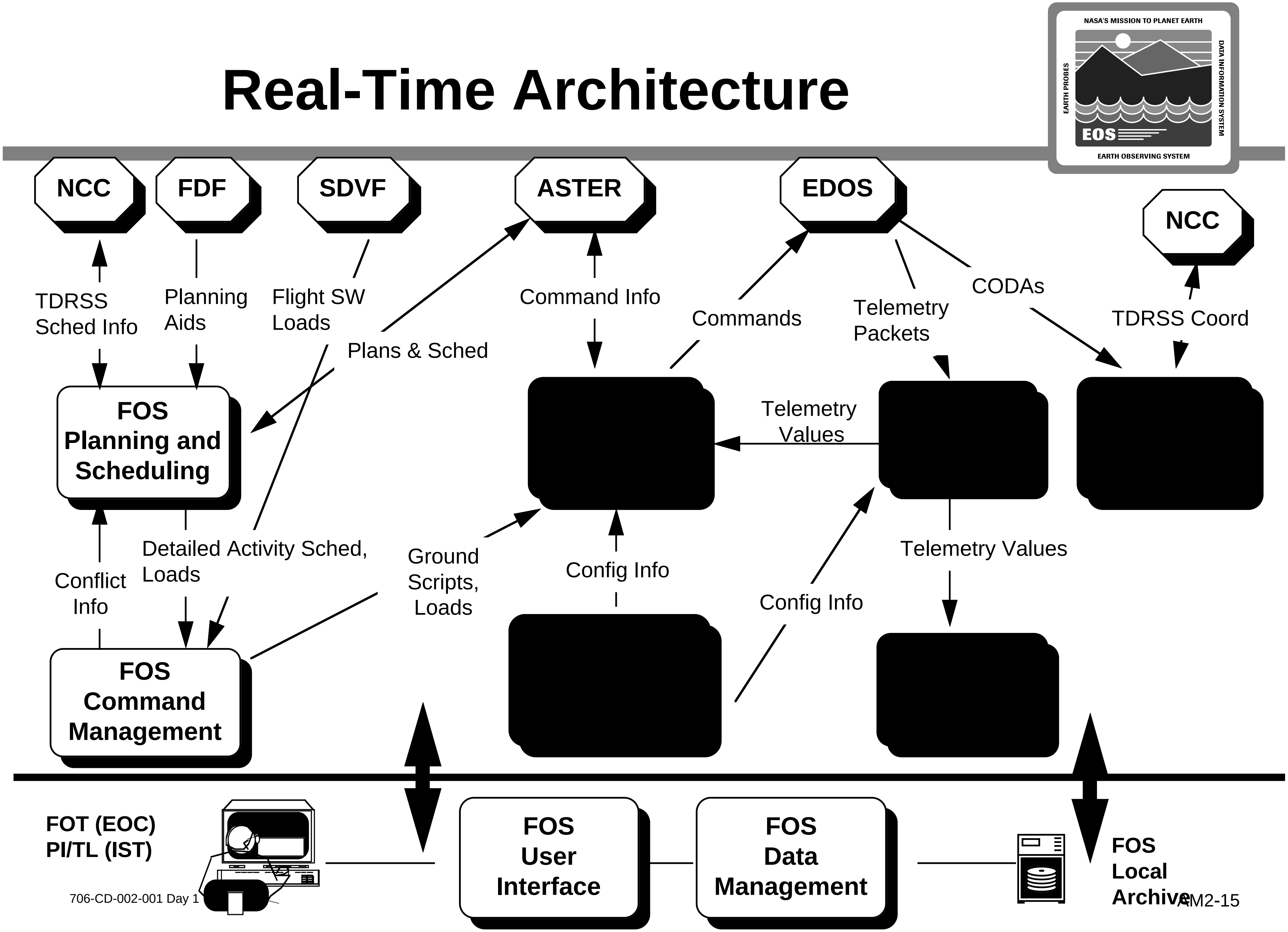
Final Scheduling

- Scheduling of ground activities
- Generates integrated conflict free detailed activity schedule

Command Load Generation

- Generation of ground scripts and ATC loads

Real-Time Architecture



Real-Time Architecture

EOC and NCC Configuration Requests

- **EOC processes user requests to establish logical string for a real-time contact**
- **NCC receives requests for space network configuration change from the EOC, if required**

Command Uplink and Verification

- **Validate, build, and transmit commands to S/C via EDOS**
- **Verify commands and command loads from CLCWs and housekeeping telemetry**

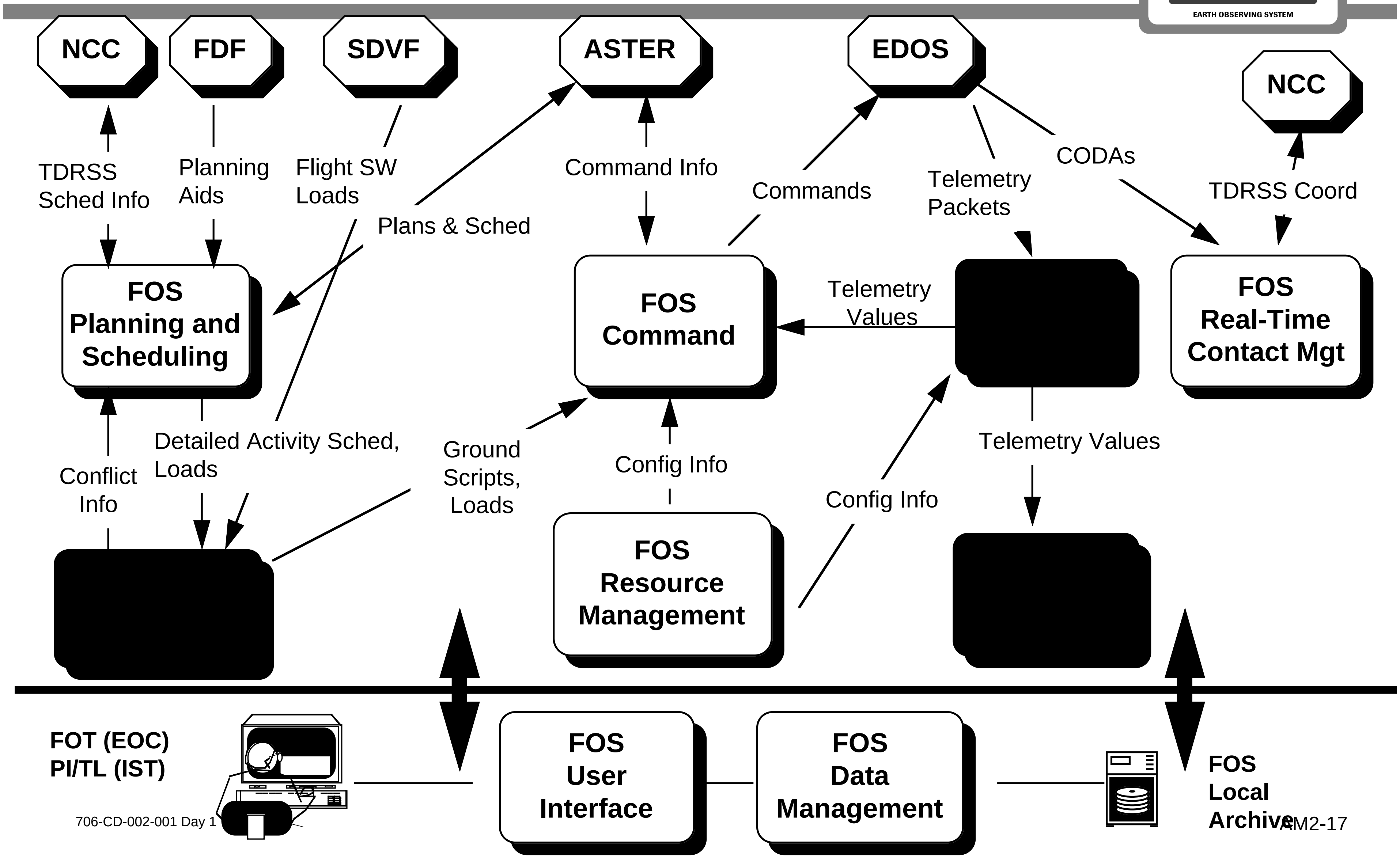
Telemetry Monitoring

- **Receive and process housekeeping telemetry from S/C via EDOS**
- **Display telemetry data and identify limit violations**

Real-time Analysis

- **SSR Management and Clock Correlation**

Off-Line Architecture



Off-Line Architecture

Routine Operations

- Analyze real-time and historical telemetry
- Assess spacecraft subsystem and instrument performance via plots, reports, statistical analysis, and trend analysis

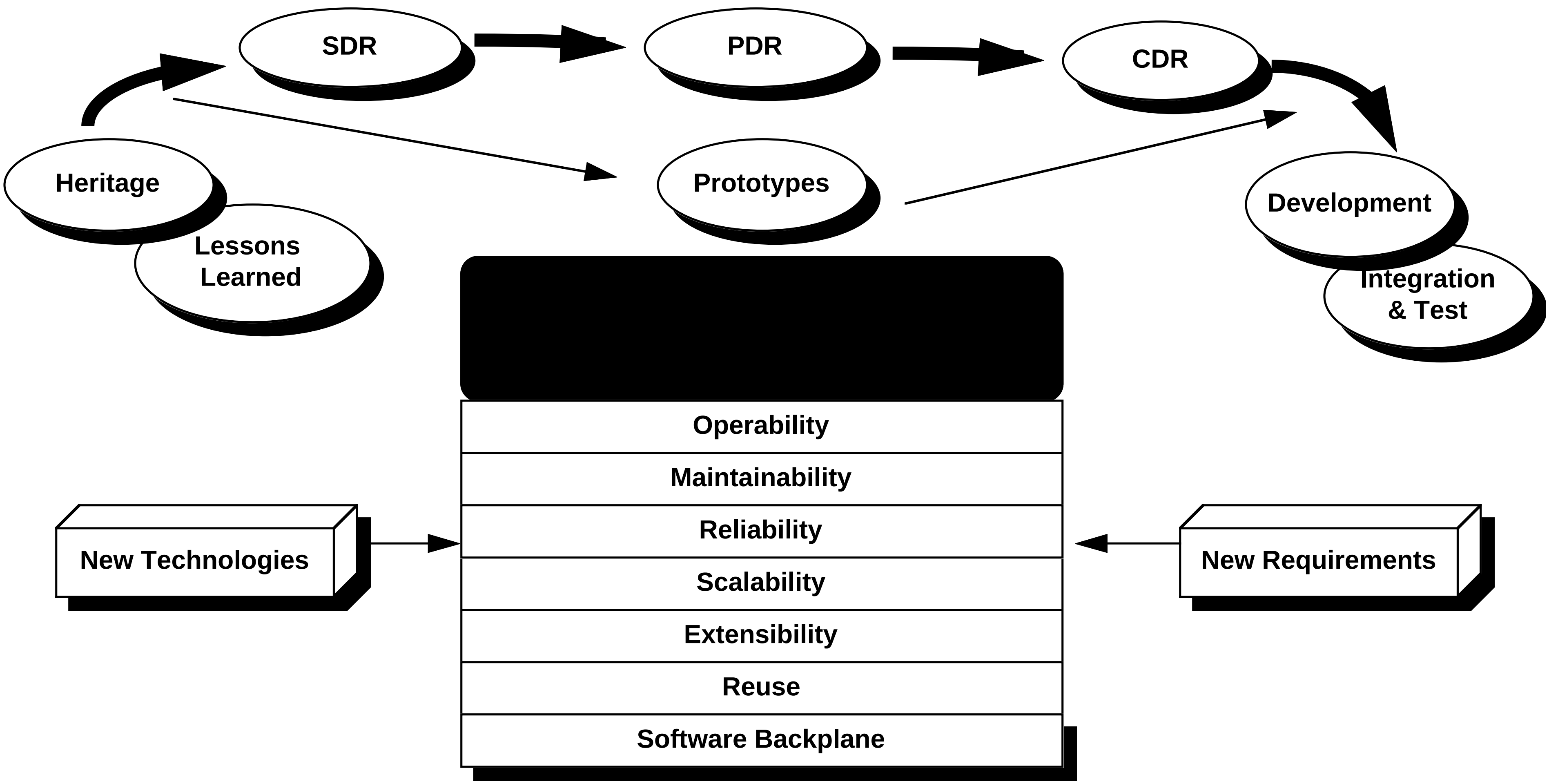
Anomaly Investigations

- Identification of resource degradation through routine operations analysis
- Use analysis tools to assist in determining scope of the problem and determine corrective action
- If time critical, then corrective action implemented during next available contact
- If not time critical, then corrective action implemented into scheduling operations

Load Management

- Production of formatted load dumps and dump comparisons

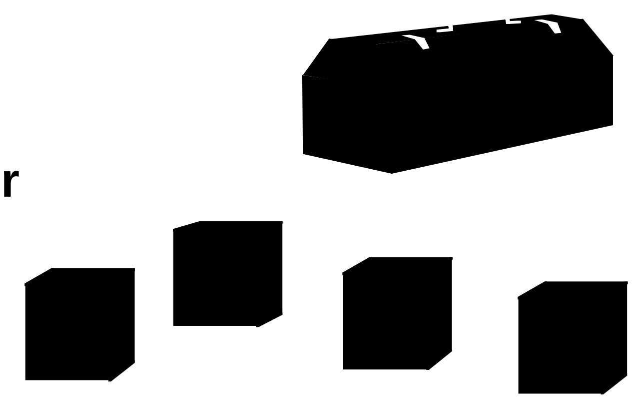
FOS Architectural Goals



System Architecture Design Objectives

1) Develop Building Blocks

- Design to reuse control center building blocks for future missions
- Design enables optimal support for technology insertion and new mission requirements
- Utilize object oriented methodology to maximize building block development
- Examples:
  - Planning and Scheduling: Resource Model
  - Resource Management: String Manager
  - Telemetry: Decom Engine
  - Telemetry: Parameter Server
  - Data Management: Queue Manager



System Architecture Design Objectives

2) Develop an End-to-end Eystem Eolution

- Link building blocks together to form loosely coupled, highly cohesive system
- Links in system provided through data products
- Example:

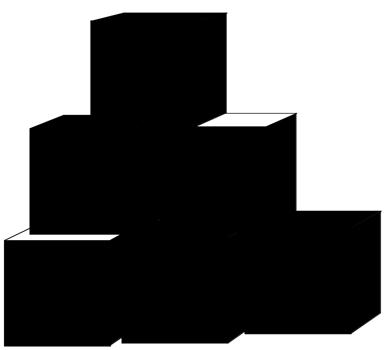
Planning and Scheduling:
Resource Model
Timeline

Activities

Command Management:
Ground Script Generation
ATC Load Generation

Ground Script

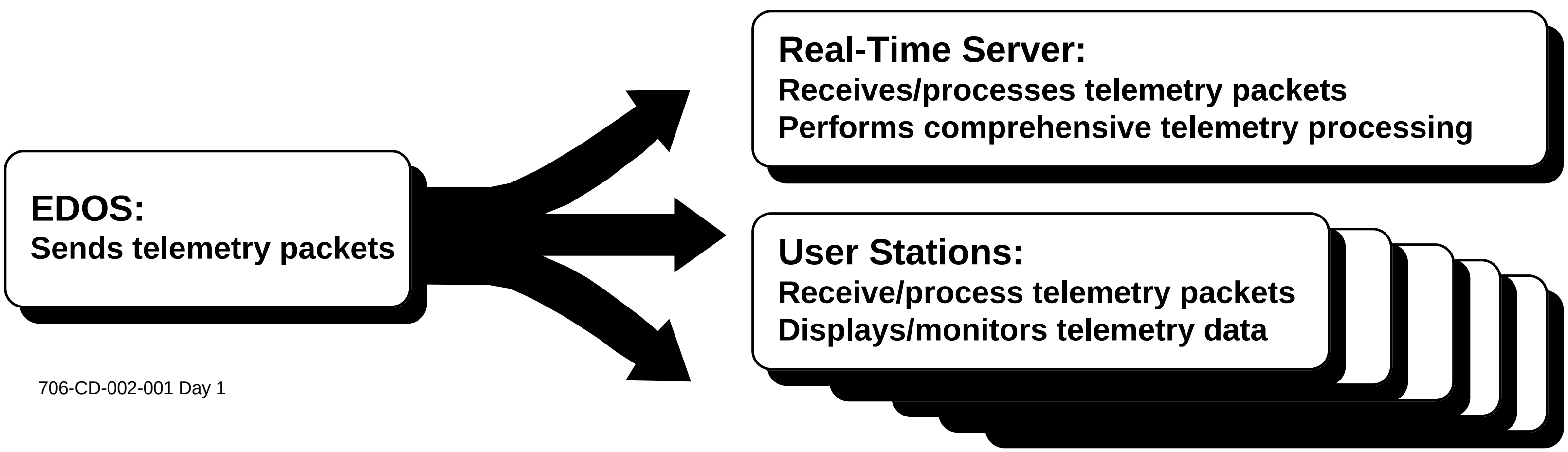
Command:
Ground Script Controller
Command Validation and Build
Command Uplink and Verification



System Architecture Design Objectives

3) Scalable Architecture

- Definition
  - Ability to increase/decrease the system size based on mission requirements
  - e.g., number of computers, users, tools
- Example: Multicasting
  - Reduces server load and network bandwidth utilization
  - New hosts are added to the system without increasing server load
  - Sending node sends only one message



System Architecture Design Objectives

4) Extensible Architecture

- Definition
  - Add new functionality to the system
  - Adapt existing system to new or changing requirements
- Example:
  - IST design provided the solution to the FOS interface for 3 different interfaces

    FDF, SDVF, and ASTER

[redacted]

EDOS
Multicast Packets

Telemetry
Decom Engine

Telemetry:
Parameter Server

Parameter Server clients:
FUI: Display
Analysis: SSR Management
Command: Execution Verification

FDF:
Real-Time Attitude Display

System Architecture Design Objectives

4) Extensible Architecture (cont.,)

[redacted]

ASTER ICC
Produce ODS, STS
Review schedules (from EOC)

IST Toolkit

Planning & Scheduling:
ASTER Interface
- Receive ODS, STS
- Send EOC schedules
Timeline tool
What-If Analysis function

FUI:
Procedure Builder
Events Display
Command Monitor tool

[redacted]

SCF
Generate Microprocessor Load
Analyze Microprocessor Dump

SDVF
Generate Flight Software Load
Analyze Flight Software Dump

IST Toolkit

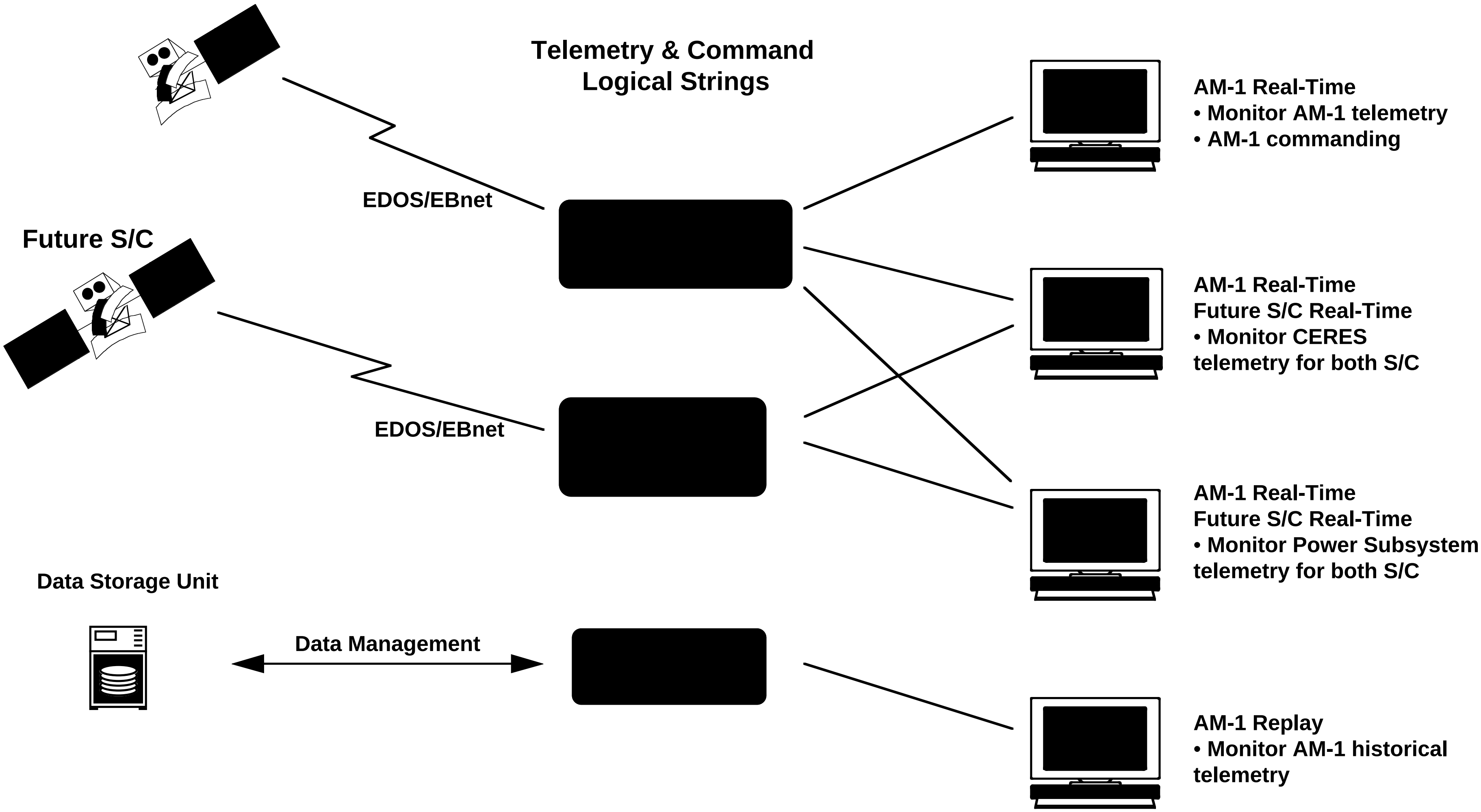
FUI:
Load Manager tool
Command Mgt:
Load Catalog
S/C Model: Dump format
Planning & Scheduling:
Load Scheduler
Timeline tool

System Architecture Design Objectives

4) Operational Flexibility

- Facilitate operational efficiency and flexibility in assigning staff personnel and tasks
- Enable FOT/IOT to run all FOS application from any workstation
- Example: Logical Strings
  - Use of Logical Strings enables the FOS to support additional processing configurations independent of hardware
  - Logical String is a collection of FOS processing resources that support a specific function (real-time contact, simulation, replay)
  - Logical Strings enable an operator to monitor data from multiple sources on the same display

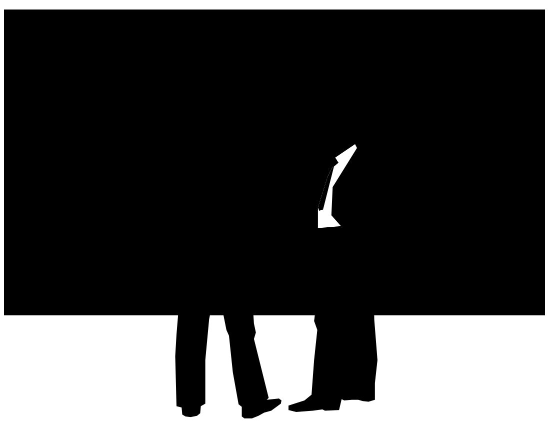
Logical String



System Architecture Design Objectives

6) Provide Global Visibility to the Users

- Examples
  - IST: remote ‘window into the EOC’ for PI/TL, IOT
  - Planning and Scheduling: Timeline
  - Resource Management: Logical String configurations
  - User Interface: Command Control window
  - Data Management: Analysis request queue
  - Analysis: Analysis Farm resources



System Architecture Design Objectives

7) Maximize Distribution of Processing

- Example: Planning and Scheduling
  - Scheduling distributed to the User Stations and ISTs
  - IOTs can schedule activities for their instruments
  - Global visibility provide via the Timeline tool
- Examples:
  - Analysis: Analysis Farm
  - Telemetry: Decom distribution to workstations

System Architecture Design Objectives

8) Provide User Customization Capabilities (operational)

- Examples:
  - User defined display pages and rooms
  - Procedure Builder tool
  - Selective decommutation and tailored telemetry processing
  - Standing Orders
  - Carry-Out products
  - Custom Report builder

System Architecture Design Objectives

9) Provide a System With High RMA

- Ensure no single point-of-failure
  - Provide redundant hardware components
    - Multiple servers
    - RAID
  - Provide redundant software components
    - Primary and Backup logical strings on different Real-Time Servers
    - Software monitoring of critical real-time software
- Promote failure recovery through software switches
  - 1 minute restoration of real-time processing

System Architecture Design Objectives

10) Provide an Evolvable Architecture that is Conducive to Increased Automation

- Increase operational efficiencies
  - Triggers based on operational events (e.g., after receipt of back-orbit telemetry)
  - Standing Orders facilitate data reduction for the FOT/IOT
- Provide automation framework
  - Example:

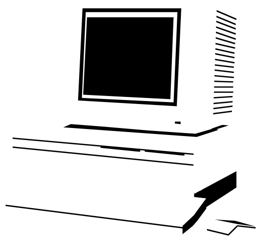
    Ground script

    Command requests based on SSR Management models

    Decision Support System

    Failure recovery through logical strings (primary and backup)

    Remote monitoring via IST



System Architecture Summary

Define an Evolvable Architecture that Facilitates Reduced Life Cycle Costs

- **Single FOS can support multiple missions concurrently**
  - **Economies of scale for FOT staffing**
  - **Future missions can use existing hardware**
    - **Logical strings used instead of physical strings**
    - **Reduces hardware maintenance costs**
- **Designed for reuse**
  - **Building blocks encapsulate functionality**
  - **New capabilities inherited from existing design**
  - **New software focused on mission customization requirements**
- **Database-driven system**
  - **Facilitates ability to add, modify, and adapt functions**

System Architecture Summary

- Platform Independence
  - Flexibility in selecting hardware based on cost/performance trades
- Improve operational efficiencies
  - Provide automation framework to facilitate increased automation in the future
  - Streamline operational tasks